computers

Article

Advancing Skin Cancer Prediction Using Ensemble Models

Priya Natha * and Pothuraju RajaRajeswari

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur 522302, Andhra Pradesh, India; rajilikhitha@kluniversity.in
* Correspondence: nathapriya@kluniversity.in

Abstract: There are many different kinds of skin cancer, and an early and precise diagnosis is crucial because skin cancer is both frequent and deadly. The key to effective treatment is accurately classifying the various skin cancers, which have unique traits. Dermoscopy and other advanced imaging techniques have enhanced early detection by providing detailed images of lesions. However, accurately interpreting these images to distinguish between benign and malignant tumors remains a difficult task. Improved predictive modeling techniques are necessary due to the frequent occurrence of erroneous and inconsistent outcomes in the present diagnostic processes. Machine learning (ML) models have become essential in the field of dermatology for the automated identification and categorization of skin cancer lesions using image data. The aim of this work is to develop improved skin cancer predictions by using ensemble models, which combine numerous machine learning approaches to maximize their combined strengths and reduce their individual shortcomings. This paper proposes a fresh and special approach for ensemble model optimization for skin cancer classification: the Max Voting method. We trained and assessed five different ensemble models using the ISIC 2018 and HAM10000 datasets: AdaBoost, CatBoost, Random Forest, Gradient Boosting, and Extra Trees. Their combined predictions enhance the overall performance with the Max Voting method. Moreover, the ensemble models were fed with feature vectors that were optimally generated from the image data by a genetic algorithm (GA). We show that, with an accuracy of 95.80%, the Max Voting approach significantly improves the predictive performance when compared to the five ensemble models individually. Obtaining the best results for F1-measure, recall, and precision, the Max Voting method turned out to be the most dependable and robust. The novel aspect of this work is that skin cancer lesions are more robustly and reliably classified using the Max Voting technique. Several pre-trained machine learning models' benefits are combined in this approach.

Keywords: multi-class skin cancer classification; ensemble models; HAM10000 dataset; random forest; gradient boosting; CatBoost; AdaBoost; extra trees; max voting method

check for updates

Citation: Natha, P.; RajaRajeswari, P. Advancing Skin Cancer Prediction Using Ensemble Models. *Computers* **2024**, *13*, 157. https://doi.org/10.3390/computers13070157

Academic Editor: Hersh Sagreiya Sagreiya

Received: 6 May 2024
Revised: 8 June 2024
Accepted: 14 June 2024
Published: 21 June 2024

1. Introduction

One common and sometimes deadly kind of cancer worldwide is skin cancer, which presents a significant challenge for medical professionals [1,2]. The effective diagnosis and precise classification of the several types of skin cancer are essential to the successful treatment of the patient. The many kinds of skin cancer present a complex challenge to predictive modeling since each form has unique traits that are essential to selecting the best course of treatment. While less common, melanoma is more likely to spread [3,4]. Sophisticated imaging techniques, such dermoscopy, have greatly improved the early detection of skin malignancies. The high-resolution images of skin lesions made possible by these methods make it easier to find features that could point to the possibility of cancer. Still, the challenge is in accurately analyzing the images to identify the particular kind of skin cancer and to distinguish between benign and malignant growths. Because there are so many different kinds of skin illnesses and the current diagnostic methods are limited, it can be difficult to identify skin cancer, occasionally resulting in inconsistent and erroneous diagnoses.

This effort is motivated by the need to raise the precision and dependability of skin cancer diagnosis. The diagnostic procedures now in use, such as biopsies and visual examinations, are time-consuming, invasive, and prone to human mistakes. The use of dermoscopic imaging has improved the early-stage condition detection capability. The subjective nature of these images still makes interpretation challenging though. Automated, non-invasive diagnostic tools that may provide accurate and consistent results are desperately needed to reduce the workload of medical staff and improve patient outcomes.

One workable answer to these problems has come from machine learning (ML) models. By using these methods, skin cancer lesions can be automatically recognized and categorized from image data. With these, the conventional diagnostic techniques [5–9] can be effectively and non-invasively replaced. There are still several areas that require attention and resolution even with the significant progress achieved in machine learning (ML) for skin cancer diagnosis. The dependency on certain datasets, the narrow classification ranges, and the complexity and understandability of sophisticated models are among the contributing elements. Many machine learning methods have been studied recently for the purpose of detecting skin cancers; each approach has advantages and disadvantages of its own, as the literature review describes. Combining many machine learning techniques, ensemble learning models have attracted attention for their ability to increase the robustness and accuracy of predictions.

The research is motivated by a number of fundamental issues. Initially, it is important to note that there exists a notable disparity in diagnoses as the prevailing diagnostic techniques frequently yield imprecise results due to human fallibility and subjectiveness. The diversity in this context can result in an incorrect diagnosis and unsuitable treatment strategies. Furthermore, several ML models now in use have a restricted classification range, concentrating exclusively on binary classification. This constraint diminishes their practical utility in real-life situations, where it is crucial to detect several categories of skin abnormalities. Furthermore, ensemble approaches, which are advanced machine learning models, typically lack interpretability. The complexity of these models can provide challenges for physicians in comprehending and placing confidence in their predictions. Finally, there is a significant dependence on certain datasets for both training and testing, which might result in overfitting. This excessive dependence diminishes the model's capacity for generalization to novel and varied data, hence constraining its efficacy in wider clinical contexts.

This work focuses on the following hypotheses: initially, we predict that ensemble learning models will surpass individual machine learning models in terms of their ability to accurately predict results and their ability to endure. Furthermore, the use of a consistent evaluation system will enhance the dependability and comparability of the findings across various studies and datasets. Ultimately, we expect that broadening the range of skin lesions included in the classification scope will improve the practical usefulness of the machine learning model in clinical environments.

Our contributions include the following:

- This study's primary goal is to examine the unique application of the Max Voting ensemble approach in order to enhance the accuracy and reliability of skin cancer lesion classifications.
- The proposed approach shows improved accuracy and robustness by combining the strengths of many pre-trained ML models. These models include Random Forest, Gradient Boosting, AdaBoost, CatBoost, and Extra Trees.
- A genetic algorithm (GA) generated the best feature vectors from a set of images. More complex ensemble learning classification techniques utilize these vectors. We assess these models using several measures, including accuracy, F1-score, recall, and precision.

The arrangement of the paper is as follows: Section 2 presents a literature review. Section 3 covers the details of the proposed method. Section 4 provides the simulation results and discussion details. Section 5 closes with the concluding remarks.

2. Literature Review

The traditional methods of identifying skin cancer, such as visual inspection and biopsy, are associated with subjectivity, require extensive and time-consuming procedures, and may exhibit significant differences amongst observers. In light of these aspects, there is a rising interest in using ML technologies to make skin cancer diagnosis more accurate and useful. This literature review introduces various ML studies on skin cancer classification in Table 1.

The study in [10] used a K-nearest neighbor (KNN) model to classify melanoma and seborrhoeic nevi-keratoses into two distinct groups. The KNN model achieved an accuracy of 85%, demonstrating its potential for distinguishing between these two types of skin lesions. However, the KNN model's performance can be sensitive to the choice of distance metric and the value of k, which may limit its generalizability across different datasets.

In [11], using a support vector machine (SVM) model, dermatological images were classified as melanoma or non-melanoma, achieving an accuracy of 90%. SVMs are powerful classifiers, but they require the careful tuning of hyper-parameters such as the kernel type and regularization parameter. Additionally, SVMs can be computationally expensive, especially for large datasets.

The authors of [12] proposed a sparse kernel representation (SKR)-based method to classify and segment skin lesions. In addition to multi-class classification (melanoma, basal cell carcinoma, and nevi), they executed binary classification (melanoma/normal), achieving respective accuracies of 87% and 92%. While this method appears to be promising, sparse kernel representation methods can be difficult to implement and may require significant computational resources.

In [13], the authors employed a new dynamic graph cut approach to segment skin lesions, and then they classified skin diseases using a Naive Bayes probabilistic classifier, achieving an accuracy of 80%. Although Naive Bayes classifiers are simple and fast, they assume independence between the features, which is often not the case in real-world data, potentially limiting their performance.

Skin cancer classification using Convolutional Neural Networks (CNNs) is the subject of the systematic review by the authors in [14]. The review showcases the use of various CNN architectures, including ResNet, Inception, and DenseNet, in skin cancer classification, with some models achieving accuracies as high as 95%. CNNs are highly effective for image classification tasks, but they require large amounts of labeled data for training and can be computationally intensive.

Advancements in the field of ensemble learning models for biological imaging have been achieved in recent works [15,16]. The most deadly type of skin cancer, melanoma, was recently predicted with an accuracy of slightly under 97% using ensemble learning [17]. Unfortunately, these researchers relied on a single-image dataset for both training and testing; therefore, it is unclear whether the model can be applied to different datasets or forms of skin cancer.

In [18], by implementing an ensemble learning approach that merged three distinct deep learning algorithms—namely VGG16, ResUNet, and CapsNet—the researchers attained a remarkable 86% accuracy rate when analyzing the ISIC Skin Cancer Dataset. Ensemble methods generally improve the model performance by combining the strengths of different algorithms, but they can be complex to implement and interpret.

The proposed approach aims to address these limitations by incorporating several key improvements:

- Diversity and Larger Datasets: To ensure robustness and generalizability, we will train and test the proposed model on a variety of datasets, including real-world clinical data.
- Multi-class Classification: The model will identify skin lesions other than melanoma and non-melanoma.
- Hybrid Model Architecture: The proposed model combines the strengths of different ML techniques, such as GA for feature extraction and ensemble learning for accuracy, to increase performance.

- Standardized Evaluation Metrics: The suggested methodology would use standardized procedures and metrics to make the study comparisons more uniform and meaningful.

Table 1. Summary of ML studies for skin cancer classification.

Study	ML Technique	Classification Task	Accuracy (%)	Demerits
[10]	KNN	Melanoma vs. Seborrhoeic Nevi-keratoses	85	Sensitive to distance metric and *k* value.
[11]	SVM	Melanoma vs. Non-Melanoma	90	Requires careful tuning, computationally expensive.
[12]	SKR	Multi-class, Binary (Melanoma/Normal).	87 (Multi-class), 92 (Binary)	Complex to implement
[13]	Naive Bayes	Skin Disease Classification	80	Assumes feature independence, may limit performance.
[14]	CNNs	Systematic Review	Up to 95	Requires large labeled datasets, computationally intensive.
[15,16]	Ensemble Learning	Biological Imaging	Not specified	Complexity in implementation and interpretation.
[17]	Ensemble Learning	Melanoma Prediction	97	Single dataset reliance, generalizability concerns.
[18]	VGG16, ResUNet, CapsNet	ISIC Skin Cancer Dataset	86	Complex ensemble approach, interpretability challenges.

3. Proposed Method

The methodology involves the combination of various pre-trained ensemble models, which ultimately results in an improvement in the overall classification performance.

3.1. Abstract Model View of the Proposed Method

The fundamental purpose of this study is to evaluate if the Max Voting model that has been suggested is capable of distinguishing between multi-class skin lesions with greater accuracy compared to specific models. The abstract model view is also shown in Figure 1.

We have listed the key steps in our proposed technique below:

1. Selecting a Proper Dataset: Choose an extensive dataset of skin lesion images, which includes seven different types of skin cancer lesions. For this investigation, we used the HAM10000 Skin Cancer Dataset [19].
2. Image Preprocessing: The following stage, picture preprocessing, involves enhancing the quality and analytical applicability of raw images through various transformations.
3. Optimal Feature Set Selection: Using a genetic algorithm (GA), the optimal feature set selection of an image set is produced.
4. Pre-trained Ensemble Models: Select a variety of pre-trained ensemble learning models for multi-class skin cancer classification, including Random Forest (RF), Gradient Boosting (GB), AdaBoost (AB), CatBoost (CB), and Extra Trees (ET).
5. Max Voting Method: Use the Maximum Voting technique to aggregate the predictions from each model separately. According to the majority vote taken by the predictions produced by the different models, this method produces predictions for each image.
6. Performance Evaluation: Evaluate the performance of the ensemble models on the test set by utilizing metrics such as confusion matrices, F1-score, accuracy, precision, and recall.

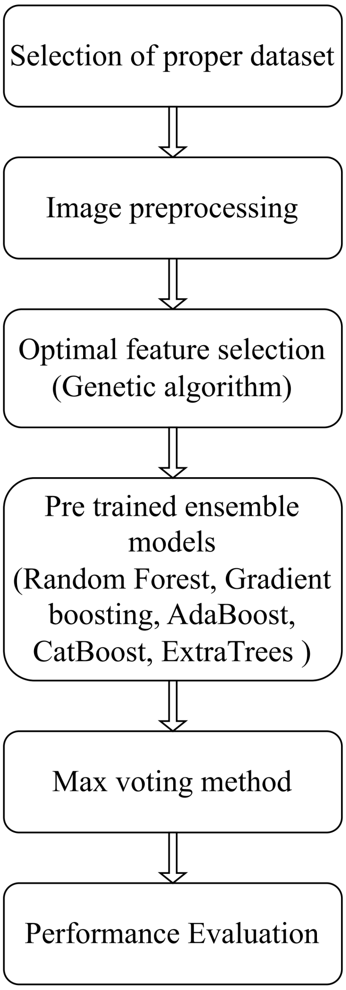

Selection of proper dataset
Image preprocessing
Optimal feature selection (Genetic algorithm)
Pre trained ensemble models (Random Forest, Gradient boosting, AdaBoost, CatBoost, ExtraTrees)
Max voting method
Performance Evaluation

Figure 1. Abstract model view.

3.2. Comprehensive View of the Suggested Model

There are multiple phases in the proposed model. Choosing an appropriate dataset that represents all seven forms of skin cancer lesions is the first step in the detection procedure. Other stages include preprocessing the image sets and extracting the ideal features using a genetic algorithm (GA). After the optimal feature set has been obtained, it is then used as input for pre-trained models, such as Random Forest (RF), AdaBoost (AB), Gradient Boosting (GB), Extra Trees (ET), and CatBoost (CB). Finally, classification is achieved by merging the outputs of individual pre-trained models using the suggested Max Voting model. We employ ensemble learning in our models to improve performance while decreasing computation time. The workflow diagram is shown in Figure 2. Algorithm 1 describes the steps of a workflow diagram.

Overfitting Problem: In order to mitigate possible issues of overfitting and dependent validation in machine learning models, many solutions can be implemented. Overfitting is the phenomenon where a model not only learns the underlying patterns in the training data but also picks up on the noise and specific features that are not useful for making predictions on new unknown data. The strategies we employed in our work are as follows:

- Cross-validation is a crucial strategy for reducing overfitting. K-fold cross-validation allows us to guarantee that the model's performance does not change when it is used on different data subsets.
- Using regularization methods like L1 and L2 can help to reduce the risk of overfitting and minimize a model's complexity. By including a regularization term in the loss

function, these techniques encourage the model to have lower weights and reduce the impact of unimportant features.

- One other method to reduce overfitting is early stopping. Using this method, the model's performance on a validation set is tracked during training and the training process is stopped when the performance stops improving. This helps keep the model from overly optimizing the training data.
- Tree-based models like Random Forest and Gradient Boosting can be pruned to remove irrelevant components that do not really affect the broad generalization of the data. This lessens the model's complexity and enhances its capacity for predicting new data.
- Ensemble techniques, which combine the predictions of several models, can help to reduce the variance related to individual models, hence reducing overfitting. Because ensemble methods combine the findings of several models, they improve the accuracy and resilience of predictions.

Dependent validation problems occur when the validation data are not completely independent from the training data, resulting in an overestimation of the model's performance. In order to tackle this issue, it is essential to appropriately divide the data. By ensuring the meticulous division of data into training, validation, and test sets, we can ensure that the validation and test sets are entirely autonomous from the training set. Creating and maintaining a distinct hold-out validation set, which is not utilized throughout the model training process, can offer an impartial evaluation of the model's performance. This enables a separate assessment of the model's capacity to generalize.

Algorithm 1 Proposed Method for Skin Cancer Lesions Classification

```
Input: Select pre-trained ensemble models (selectedensembleModels)
Select skin lesion images (input Images)
Equivalent Target labels (TargetLabels)
Output: Each individual input image is predicted by an ensemble model (ensemble predictions)

Step 1: Ensemble Model selection
selected models $\Longleftarrow$ [Random Forest, Gradient Boosting, AdaBoost, CatBoost, Extra Trees]

Step 2: Image preprocessing
while InputImages $\neq$ eachImage do
Preprocessed Image $\Longleftarrow$ [Resizing and Cropping, Rescale, Normalization, Color Correction, Noise Reduction]

Step 3: Optimal feature extraction of Images
Optimal feature extraction of an Image $\Longleftarrow$ [Genetic algorithm]

Step 4: Load Ensemble Pre-trained Models
Loaded Models $\Longleftarrow$ load Models (selected Models)

Step 5: Generate Predictions
while eachloadedmodel $\neq$ LoadedModels do
ModelPrediction $\Longleftarrow$ PredictImage (LoadedModel, PreprocessedImage)

Step 6: Max Voting method
while eachImagePrediction $\neq$ setof predictions do
MajorityVote $\Longleftarrow$ calculateMajorityVote (imagePrediction)

Step 7: Evaluate Models' Performance
Accuracy $\Longleftarrow$ evaluateAccuracy(ensemblePredictions, TargetLabels)
```

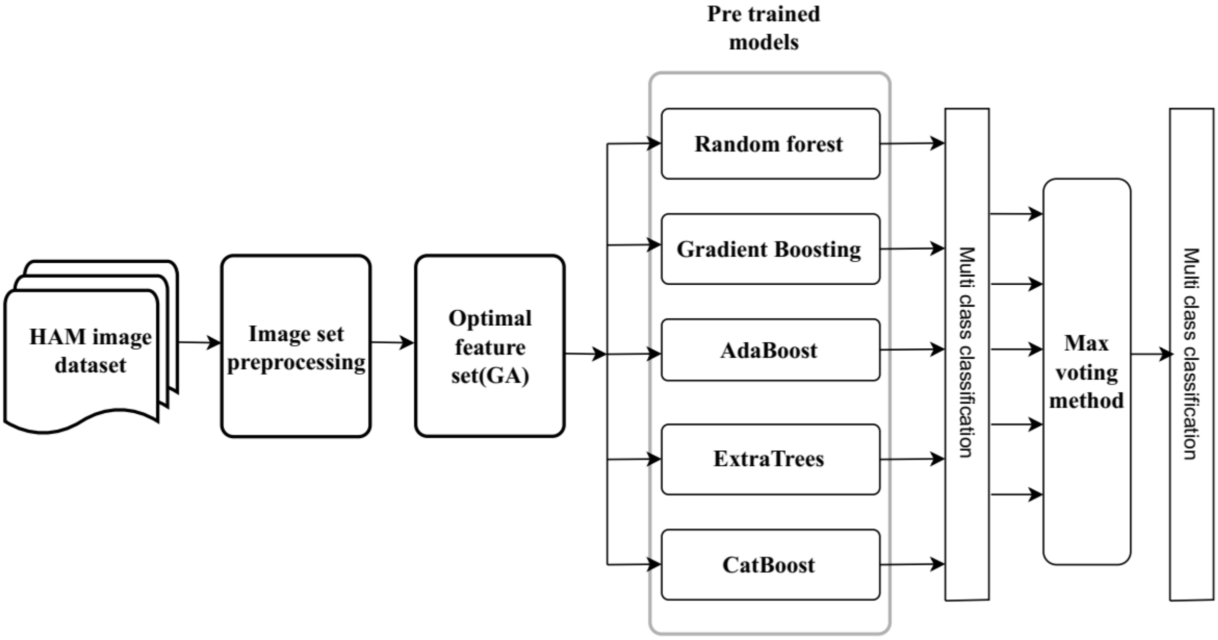


Figure 2. Workflow diagram.

3.3. DATASET

Used the HAM10000 dataset ('Human Against Machine with 10,000 skin cancer lesion images') [19], which contains 10015 dermatoscopic images and seven classes, including actinic keratosis (akiec) (327), basal cell carcinoma (bcc) (514), benign keratosis (bkl) (1099), dermatofollioma (df) (155), melanocytic nevi (nv) (6705), melanoma (mel) (1113), and vascular skin lesions (vasc) (142). The seven types of sample images are shown in Figure 3. To better understand the distribution of images within this dataset, refer to Table 2 for an image dataset distribution. Seven distinct types of skin cancer images are assigned to the seven class numbers. There is an imbalance in the dataset as the number of images belonging to the melanocytic nevi class is greater than the number of images from other classes. In order to solve this, the frequency of each class was increased based on the data augmentation method [20,21], resulting in an equal amount of approximately 3200 images for each class in the modified image dataset. As a result, the dataset became more precise and balanced. During the evaluation phase, we utilized this validation dataset to validate the various sets for the skin lesion dataset. The class numbers assigned to the lesions are actinic keratosis (class-0), basal cell carcinoma (class-1), keratosis (class-2), dermatofibroma (class-3), melanocytic nevi (class-4), melanoma (class-5), and vascular skin lesions (class-6).

Table 2. Image dataset distribution.

Name of Lesion	Total Images (10,015)	Distribution after Data Augmentation
Actinic Keratosis	327	3235
Basal cell Carcinoma	514	3283
Benign keratosis	1099	3170
Dermatofibroma	155	3252
Melanocytic nevi	6705	3179
Melanoma	1113	3195
Vascular Lesion	142	3189

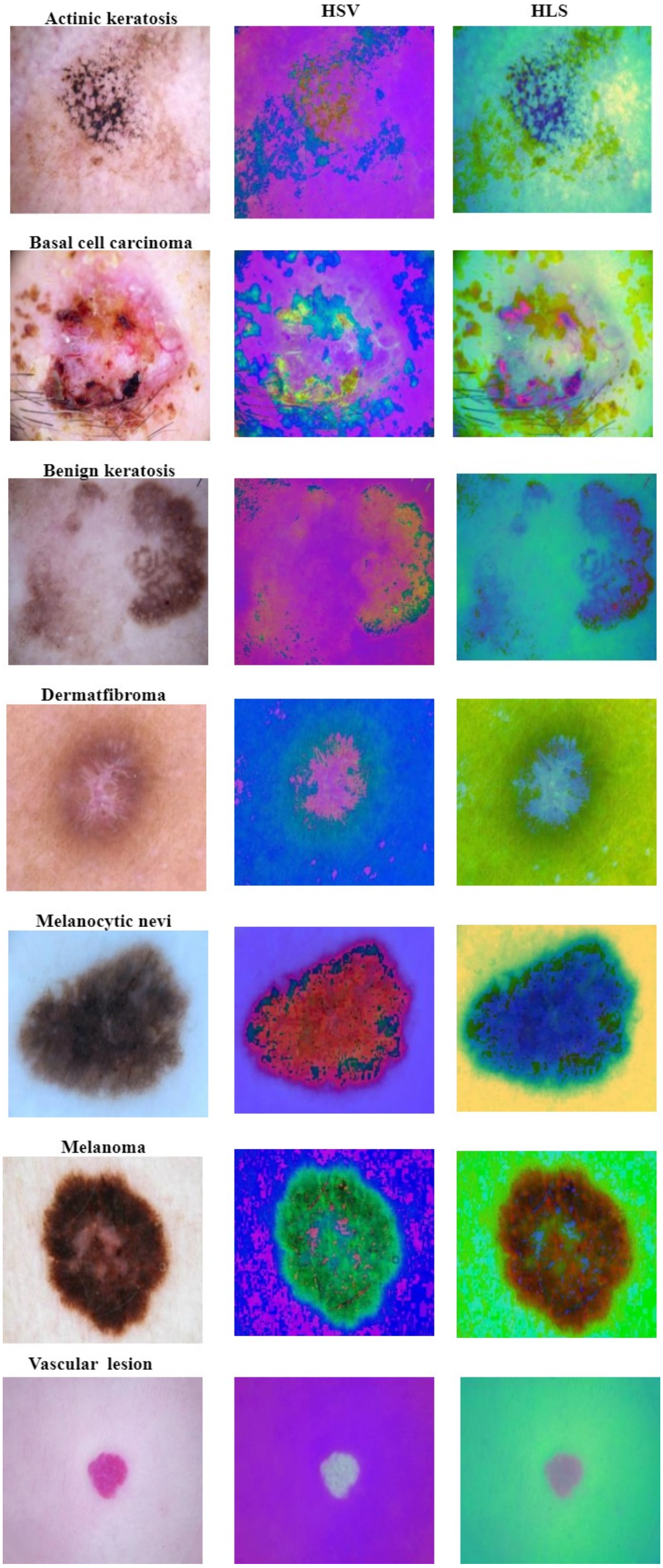


Figure 3. Seven types of images and their HSV and HSL color spaces.

3.4. Image Preprocessing

Image preprocessing is an important part of analyzing skin cancer images because it makes the images better and brings out the features that are needed for detection. These are the standard preprocessing procedures for images of skin cancer.

- Resizing: Resizing refers to altering the size of an image by modifying its dimensions. Within this particular context, the initial images of skin cancer are resized to a consistent dimension of 256 × 256 pixels. Ensuring uniformity in the image dimensions is essential for maintaining consistency in the analysis and processing procedures. It guarantees uniform dimensions for all images, hence simplifying computational tasks like feature extraction and model training.
- Color Space Conversion: Converting images to grayscale improves the study by eliminating color information and concentrating exclusively on the light's intensity. Since this conversion enhances the visibility of features linked to texture and shape in skin lesions, it is particularly useful in medical image analysis. Images in grayscale only exist in levels of gray, where the values of the individual pixels represent the intensity of the light, from black to white.
- Contrast Enhancement: Contrast enhancement refers to the application of techniques that increase the intensity difference between different regions in an image in order to improve the visibility of its characteristics. One often-used technique for increasing contrast is histogram equalization. It changes how pixels' intensity values are arranged in an image, extending the range of intensities to include the whole grayscale range. The contrast in the image is much increased by this technique, which makes the intricacies inside the skin lesions more visible than the background.
- Denoising: Unwanted noise, which can originate from a variety of causes, including ambient conditions or sensor errors, is removed from an image by a technique called denoising. Preserving the integrity of lesion boundaries and raising the accuracy of later analysis depend on noise reduction in skin cancer images. One often-used method for reducing noise in images is the median filter. The procedure is replacing the median value of the surrounding pixels for every pixel in the image. This filter works well to improve contrast-enhanced images' clarity by reducing noise while maintaining edge details.
- Edge Detection: A key stage in image processing, edge detection is used especially to find object boundaries. The diagnosis of skin cancer depends on precise definition of the boundaries of lesions. One often used technique for detecting major changes in intensity in an image to identify edges is the Canny edge detector. The way this system works is by image smoothing, edge magnitude and direction computation using gradients, and edge detection and connection using hysteresis thresholding.
- Data Augmentation: A method known as data augmentation modifies current images to deliberately expand and diversify a collection. Among the many possibilities for these adjustments are rotation, scaling, flipping, and cropping. The analysis of skin cancer images requires data augmentation since it improves the flexibility and applicability of machine learning models to a variety of situations. The model enhances its ability to identify a wide range of patterns and features by producing more variants of skin lesion images, which enhances its performance on unknown data.

3.5. Feature Extraction

By converting RGB images of skin lesions to other color spaces [22,23] such as CMYK (Cyan, Magenta, Yellow, and Key/Black), HSL (Hue, Saturation, and Lightness), XYZ Color Space, and HSV (Hue, Saturation, and Value), it becomes easier to extract and analyze color-related and textural data. These aspects are essential for identifying and classifying different types of skin cancer.

In the context of skin cancer lesions, feature extraction refers to the process of evaluating images of the skin in order to recognize and quantify significant features or characteristics that are capable of distinguishing between different types of lesions. There are usually two extracted features:

(i) Color Feature Extraction:

- Color histograms: They show the lesion's color distribution by calculating the proportion of pixels that belong to each color bin. Histograms can be generated for each channel in RGB, HSV, or other color spaces to capture the overall color distribution.
- Color Moments:
    - Mean (First Moment): Displays the dominant color and the average color intensity for each channel during the lesion.
    - Standard Deviation (Second Moment): Calculates color variability and displays the degree of uniformity in the color distribution.
    - Skewness (Third Moment): Determines how symmetrically the color distribution is, which may point to the existence of uneven pigmentation.

(ii) Texture Feature Extraction:

Given that different textual characteristics may be observed in different kinds of benign lesions and skin malignancies, the use of texture features in the analysis and classification of skin cancer lesions from images is of great relevance [24–26]. To investigate an additional complex technique for texture measurement: Gray-Level Co-occurrence Matrices (GLCM) [27] are matrices that show, from their gray-level values, the spatial association between pairs of pixels in an image. A potent statistical instrument for texture analysis, the GLCM captures the spatial relationship in an image between pairs of pixels at a given distance and direction.

Step 1: Constructing the GLCM

A GLCM is constructed by considering pairs of pixels at a certain distance b and direction c from each other. For a grayscale image with l gray levels, the GLCM will be a $l \times l$ matrix where each element $P(u, v|b, c)$ represents the frequency of pixel pairs with intensities u and v.

1. Initialization: Start with a zero matrix P of size $l \times l$.
2. Pixel Pair Counting: For each pixel in the image, consider another pixel at distance b in direction c. If the intensity of the first pixel is u and the second pixel is v, increment $P(u, v)$ by 1.
3. Normalization: After counting all relevant pixel pairs, normalize the matrix so that the sum of its elements equals 1. This converts frequency counts into probabilities.

$$P_{norm}(u, v|b, c) = \frac{P(u, v|b, c)}{\sum_{u=0}^{l-1} \sum_{v=0}^{l-1} P(u, v|b, c)}. \tag{1}$$

Step 2: Computing Texture Features from GLCM

After constructing the GLCM, several statistical measures can be extracted to describe the texture of the image:

1. Contrast measures the local variations in the GLCM.

$$\text{Contrast} = \sum_{u=0}^{l-1} \sum_{v=0}^{l-1} (u - v)^2 P_{norm}(u, v|b, c). \tag{2}$$

2. Energy (or Angular Second Moment) reflects the uniformity of the GLCM.

$$\text{Energy} = \sum_{u=0}^{l-1} \sum_{v=0}^{l-1} P_{norm}(u, v|b, c)^2. \tag{3}$$

3. Correlation indicates how correlated a pixel is to its neighbor over the entire image.

$$\text{Correlation} = \frac{\sum_{u=0}^{l-1}\sum_{v=0}^{l-1}(u-\mu_u)(v-\mu_v)P_{norm}(u,v|b,c)}{\sigma_u\sigma_v}. \tag{4}$$

Here, μ_u and μ_v are the means and σ_u and σ_v are the standard deviations of the row and column sums of the GLCM, respectively.

4. Homogeneity (or Inverse Difference Moment) shows the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

$$\text{Homogeneity} = \sum_{u=0}^{l-1}\sum_{v=0}^{l-1}\frac{P_{norm}(u,v|b,c)}{1+(u-v)^2}. \tag{5}$$

5. Entropy measures the randomness in the GLCM.

$$\text{Entropy} = -\sum_{u=0}^{l-1}\sum_{v=0}^{l-1}P_{norm}(u,v|b,c)\log(P_{norm}(u,v|b,c)). \tag{6}$$

These features collectively provide a comprehensive view of the texture of the image, capturing aspects like uniformity, complexity, and the relationship between pixel intensities. They are widely used in image analysis for various applications, including medical imaging, where they help in the classification and diagnosis of diseases by analyzing patterns in tissue and organ images.

3.6. Feature Selection Optimization

The goal of applying genetic algorithm (GA) [28] is to identify the optimal subset of the features that will allow us to classify the images accurately while using a small set of features. The steps of GA are shown below. Table 3 displays the GA method's key parameters and their values for the study.

- Step 1: Define the GA Components:
  - Chromosome Representation: In feature selection tasks, chromosomes are usually shown using a binary string, where each bit in the string represents a different feature from the set of features, where the value of each position indicates whether a particular feature is selected (1) or not selected (0).
  - Fitness Function: By utilizing this fitness function, the GA is directed to find the best subset of features that both simplify the model and achieve high classification accuracy.
- Step 2: Initialize Population: Start with a randomly generated population of chromosomes.
- Step 3: GA Operations:
  - Selection: To choose parent chromosomes for breeding, use tournament selection.
  - Crossover: To produce offspring, use a single-point crossover. For instance, the first 10 features of one parent are combined with the remaining features of the other parent if the crossover point is at position 10.
  - Mutation: Introduce variety into the offspring by applying mutation to them with a given probability (mutation rate).
  - Replacement: Incorporate the new offspring into the population to replace the least-perfect chromosomes.
- Step 4: Iterate: Continue selecting, crossover, mutating, and replacing until the fitness score no longer increases noticeably, or for a predetermined number of generations.
- Step 5: Evaluate the Best Solution: Select the best-performing chromosome based on the optimization function result. This chromosome stands for the ideal selection of characteristics.

Table 3. Key parameters of GA.

Key Parameters of GA	Values
Population	30 chromosomes
Fitness Function	Support Vector Machine (SVM)
Tournament Selection	size of 3
Crossover	crossover rate of 0.7
Mutation	Bit flip mutation rate of 0.01
Termination	stops early if the accuracy exceeds 95%.

Some of the important functions used in GA are elaborated on further.

(i) Chromosome Representation: Each chromosome in the population represents a potential solution to the feature selection problem and is encoded as a binary string. Let $X = [x_1, x_2, ..., x_N]$ represent a chromosome, where N is the total number of features, and x_i is a binary variable indicating the inclusion (1) or exclusion (0) of feature i in the model.

(ii) Fitness Function: The fitness function $f(X)$ quantifies the quality of a solution. For feature selection, it often combines the predictive performance of a model using the selected features with a penalty for the number of features to discourage overfitting. A typical fitness function could be

$$f(X) = \alpha \cdot \text{Accuracy}(X) - \beta \cdot \text{Count}(X). \tag{7}$$

where

- $\text{Accuracy}(X)$ quantifies the predictive accuracy or effectiveness of the classifier when using the subset of features selected by X. This could be measured through accuracy, F1-score, or area under the ROC curve (AUC).
- $\text{Count}(X)$ reflects the complexity of the model, often represented by the number of features selected (i.e., the number of 1s in X). The rationale is to penalize solutions that use more features than necessary to prevent overfitting and ensure model simplicity.
- α and β are weighting parameters that balance model accuracy and complexity.

(iii) Optimization Function: The goal is to find the subset of features X that maximize a predefined fitness function $f(X)$. The fitness function evaluates the quality of a solution, which usually involves a balance between the performance of a predictive model using the selected features and the complexity of the model.

The formal mathematical formulation of this optimization objective is

$$X^* = \arg\max_X f(X). \tag{8}$$

where

- X^* is the optimal subset of features that we aim to find.
- X represents a candidate solution, which is a subset of the available features in the dataset.
- $f(X)$ is the fitness function that quantifies the quality of the solution X.

In this formulation, the use of $\arg\max$ indicates that we seek to maximize the fitness function. In feature selection, a higher value of $f(X)$ typically implies a better subset of features, considering both the predictive power and the simplicity of the model.

3.7. Ensemble Models

As an ML technique, ensemble learning combines the predictions of several models to make the performance and robustness better. By combining the predictions from various models, the shortcomings of individual models are supposed to be lessened, improving accuracy and generalization. The research in this paper made use of the following popular types of ensemble methods.

- Bagging (Bootstrap Aggregating): Bootstrapping [29] is the process of training a model many times using different replacement-sampled portions of the training data. The ultimate forecast is acquired by classifying the forecasts of individual models through voting. A well-liked technique called Random Forest employs bagging, with decision trees serving as the foundational models.
- Boosting: The purpose of boosting [30] is to make underperforming models more accurate by assigning greater importance to instances that were incorrectly classified. To fix the misclassifications, a new weak model is trained for each iteration. The sum of the weighted predictions made by each separate model is the final prediction. Gradient boosting, AdaBoost, and CatBoost are a few examples.

3.7.1. Random Forest (RF)

Random Forest [31] is an ensemble learning algorithm that uses multiple decision trees to create robust, accurate models by reducing variance and overfitting in classification tasks. In order to train numerous decision trees, this method uses the feature matrix that is generated from the image set. A random subset of the features and data is used to train each tree, which minimizes overfitting and enhances generalization. For instance, an RF model may determine that melanoma with asymmetrical forms and irregular boundaries is extremely likely malignant. Table 4 displays the important RF technique parameters and their values employed in this study.

Table 4. Key parameters of RF.

Key Parameters of RF	Values
Number of Trees	60
Criterion	gini
Maximum Depth	30
Minimum Samples Split	2
Minimum Samples Leaf	5
Bootstrap	True

3.7.2. Gradient Boosting

When compared to Random Forest, which constructs trees simultaneously, Gradient Boosting (GB) [32] constructs trees sequentially. New trees are trained to fix mistakes made by older ones. Even when distinguishing between skin cancers that are very diverse in appearance and subtle in their differences, this iterative correction method frequently produces very accurate results. To avoid overfitting, Gradient Boosting incorporates adjustable parameters like learning rate and number of trees. Table 5 displays the important GB technique parameters and their values used in this study.

Table 5. Key parameters of GB.

Key Parameters of GB	Values
Loss Function	Multinomial logistic loss
Number of Estimators	60 trees
Learning Rate	0.1
Maximum depth	15

Table 5. *Cont.*

Key Parameters of GB	Values
Minimum Samples Split	2
Minimum Samples Leaf	5
Sub-sample	0.8

3.7.3. AdaBoost

AdaBoost (Adaptive Boosting) [33] turns a number of poor classifiers into one powerful classifier. AdaBoost (AB) builds the ensemble of classifiers iteratively, starting with a basic classifier. As it adjusts to the complexities of the data, each new classifier concentrates more on the cases that the prior classifiers misclassified. By weighing these weak learner predictions according to accuracy, the algorithm aggregates these weighted predictions to arrive at a final conclusion. This process turns these weak learners into strong classifiers. The key parameters and their values of the AB method used in the study are shown in Table 6.

Table 6. Key parameters of AB.

Key Parameters of AB	Values
Base Estimator	decision tree with depth = 1
Number of Estimators	50 weak learners
Learning Rate	0.1
Algorithm	SAMME.R
Random State	random state = 42

3.7.4. CatBoost

CatBoost (Categorical Boosting) [34] is a sophisticated ensemble learning method known for its performance with little data preprocessing and built to operate well with categorical data. CatBoost (CB) is well-suited for the subtle variations between skin cancer types because of its ability to manage intricate data patterns and feature interactions. A CatBoost model is then trained using the features that were extracted. In order to make sure the model works effectively on unknown data, it is validated using techniques like cross-validation. This is an important step, where accuracy is crucial. The key parameters and their values for the CB method used in this study are shown in Table 7.

Table 7. Key parameters of CB.

Key Parameters of CB	Values
Number of Trees	70
Number of Estimators	50 weak learners
Learning Rate	0.1
Loss Function	cross-entropy loss
Evaluation Metric	F1-score
Boosting _ type	Ordered
auto_class_weights	Balanced
l2_leaf_reg to a value	3

3.7.5. Extra Trees

An ensemble learning technique called the Extra Trees (Extremely Randomized Trees) [35] algorithm can be especially useful for classification tasks. It expands on the decision tree framework, just like other tree-based techniques, but it adds more randomness to the split selection process, which may result in models that are more resilient and have lower variation. Using the training dataset, Extra Trees (ET) constructs several decision trees. It adds additional randomization by choosing cut-points entirely at random as opposed to choosing the optimal split among a random subset of the features, as is the case with other ensemble methods like Random Forest. Additionally, it grows the trees using the entire learning sample instead of a bootstrap copy. A majority vote is used to determine the final forecast for classification tasks. The key parameters and their values of the ET method used in this study are shown in Table 8.

Table 8. Key parameters of ET.

Key Parameters of ET	Values
Number of Estimators	50 trees
Criterion	gini
Minimum Samples Split	4
Max Features	auto
Min Samples Leaf	2
Bootstrap	False
number of jobs	n_jobs = −1

3.8. Max Voting Mechanism

In this method, several distinct models use the same dataset to train. These models can belong to the same class (homogeneous ensemble) or to distinct classes (heterogeneous ensemble). Each of the trained models predicts the class label for a particular input. Next, for every input sample, the predictions made by every model are combined by the Max Voting (MV) technique [36]. The class that receives the most votes is taken into consideration as the sample's final prediction. Several techniques, including weighted voting based on model confidence, random selection, and giving priority to the vote of the more accurate model, can be used in the event of a tie. Thus, the class that receives the greatest number of votes across all models for each input sample is known as the "max vote", and this determines the final output prediction. The key parameters and their values of the MV method used in this study are shown in Table 9.

Table 9. Key parameters of MV.

Key Parameters of MV	Values
Individual Classifiers	RF, GB, AB, CB, ET
Voting	Majority-Weighted votes
random_state	42

The Max Voting decision D for an input sample in an ensemble learning context is determined by selecting the class that receives the most votes from all the models in the ensemble. Mathematically, this decision rule is formulated as follows:

$$D = \text{argmax}_{j\in\{1,2,\ldots,k\}} \sum_{i=1}^{N} \mathbb{1}(p_i = C_j). \tag{9}$$

Here is what each part of the formula represents:

- D: The final decision or prediction made by the Max Voting algorithm.
- argmax: A function that finds the argument j that maximizes the given expression.
- j: An index representing the class labels $C_1, C_2, \ldots, C_k$.

- k: Number of unique classes.
- $\sum$: The symbol that sums all N models' votes.
- N: The total number of models in the ensemble.
- $\mathbb{1}(\cdot)$: The indicator function, which returns 1 if the condition inside is true (i.e., if model i predicts class C_j) and 0 otherwise.
- p_i: The prediction made by model i for the input sample.
- C_j: The jth class label among the possible class labels.

In essence, the Max Voting decision D is the class that the majority of the models predict for the given input sample, thereby reflecting a consensus choice among the models.

The Max Voting method employs various strategies to ensure a systematically and reliably determined final prediction when multiple classes receive equal votes. One approach is weighted voting based on model confidence, where votes from different models are weighted according to their confidence levels. The class with the highest total weighted score across all models is selected as the final prediction. This method leverages the models' confidence to make more informed decisions. Another strategy is random selection, which involves selecting one of the tied classes at random and using a random tie-breaking mechanism to select the final prediction. This method ensures a decision is made even when votes are tied. The process involves prioritizing the most accurate model based on past performance, and, if a tie occurs, the most accurate model's vote determines the outcome.

4. Simulation Results and Discussion

This section describes the experimental evaluation in detail, including the experimental setup and hardware and software environment requirements.

4.1. Experimental Setup

Colab and Kaggle were the software environments employed by using CUDA version 11.2; Kaggle GPU resources comprised the hardware environment is shown in the Table 10. An 80–10–10 split was used to divide the dataset into three parts: the training set, the test set, and the validation set. The training set contains 80% of the data, while the test set and validation set each contain 10%.

Table 10. Hardware specifications of experimental setup.

Key Components of Hardware	Specifications
NVIDIA Windows Driver	546.12
NVIDIA GPU	GeForce Titan Xp
CPU Chip	Intel core i7
CPU Cores	4
Chip Speed	3.2 GHz
CPU RAM	12 GB
Disk Space	320 GB

4.2. Features analysis

To determine an optimal amount of features that might balance the performance and resource usage, a thorough feature analysis was completed. This was conducted to make sure that the performance is maintained while the system runs effectively. To detect the changes in skin cancer lesions induced by malignancy, the individual color [37–39] and texture features [40–42] were assessed for multi-class classification. There were 342 texture features and 68 color features at first, for a total of 410 features. After the features were optimized to the optimal number in each scenario using the information gathered from the features, cross-validation was carried out.

For this multi-class categorization of the lesions related to skin cancer, using the genetic algorithm (GA), we were able to determine that the ideal feature set size was 61. It was found that 23 features notably had the largest information gain while looking simply at the color features. Similarly, 38 features were shown to have the largest information gain when the texture features were taken into consideration. After combining the color and texture features, as indicated in Table 11, it was found that 61 features had the highest information gain. These features, when subjected to 10-fold cross-validation, produced the following results: an accuracy of 95.80%, precision of 95.44%, recall of 95.04%, and F1-score of 95.20%, which resulted in an 85% reduction in the total number of features. This experiment revealed that color and texture characteristics both have a major role in predicting skin cancer's precancerous phases.

Table 11. A 10-fold cross-validation was conducted on a validation set to see how well the mix of color and texture features worked in the Max Voting method.

Iteration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Iteration1	92.03	91.37	91.70	98.55
Iteration2	97.48	90.63	93.93	91.37
Iteration3	93.10	97.83	95.41	94.12
Iteration4	96.73	98.01	97.37	93.54
Iteration5	97.79	96.73	97.26	96.24
Iteration6	93.50	91.78	93.39	92.45
Iteration7	92.60	90.08	92.59	93.66
Iteration8	94.90	92.46	91.69	95.87
Iteration9	94.90	95.48	97.39	91.22
Iteration10	96.08	90.74	93.33	92.65
Average	95.80	95.04	95.44	95.20

4.3. Evaluation Parameters

In order to determine how successfully a machine learning classification model predicts categorical labels, a variety of measures are used in its evaluation [43]. The values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are produced by the confusion matrix. The values derived from the confusion matrix are used to calculate the evaluation parameters. The following are a few typical classification model evaluation parameters:

- Confusion Matrix: The table provides a concise overview of the classification model's performance. It aids in comprehending the model's performance across several classes by providing a thorough breakdown of the FN, FP, TP, and TN.
- Accuracy: This performance metric, which is the most logical one, is just the proportion of correctly predicted observations to the total number of observations. When the target classes are evenly distributed, it is helpful.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{10}$$

- Precision: Precision is the proportion of TP predictions out of all the positive predictions generated by the method. It aids in analyzing the model's ability to detect FPs.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{11}$$

- Recall (Sensitivity): A dataset's true positive cases corresponding to true positive predictions are determined by the recall function. It facilitates the comprehension of the model's capacity to identify every positive case.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{12}$$

- F1-score: The harmonic mean of precision and recall is known as the F1-score. Because it attains high values for both precision and recall, a high F1-score is an excellent indicator of your model's performance. For classes that are unbalanced, this metric is preferable to accuracy.

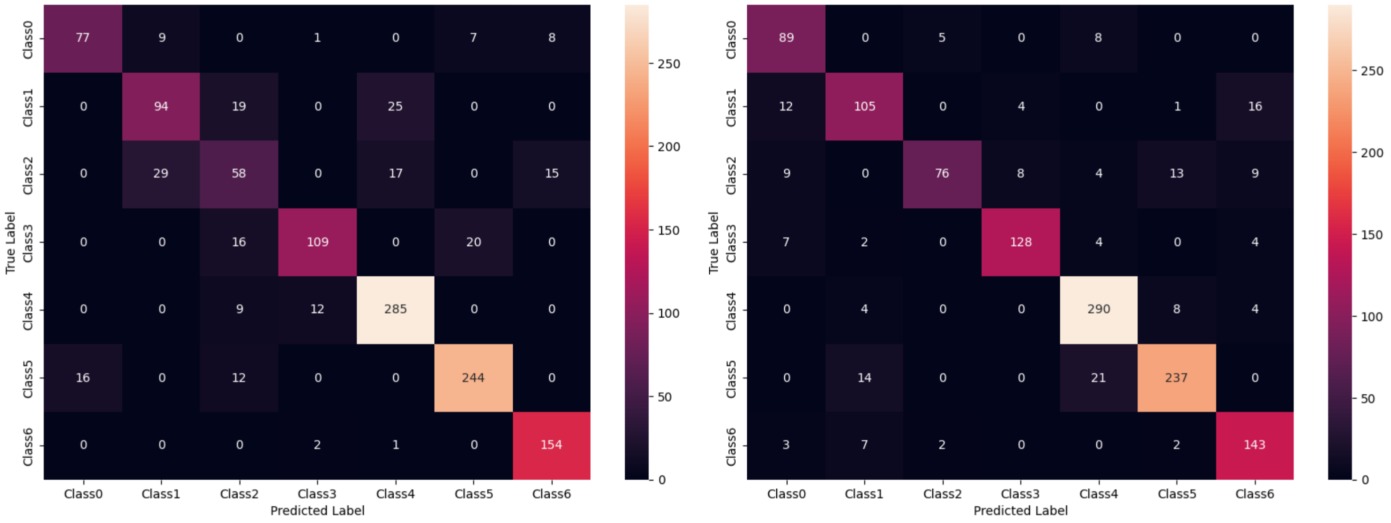
$$\text{F1-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{13}$$

4.4. Error Analysis and Confusion Matrix (CM)

A tabular representation for describing the performance of a classifier is provided by the confusion matrix (CM) [44]. The confusion matrices for six different models are shown in Figure 4. These matrices provide a combined perspective of the prediction results, differentiating between accurate and inaccurate classifications for seven classes.

To show how many accurate and inaccurate predictions there were for each class, counts are employed within the matrix. Visualizing the categorization errors and confusion that may occur during predictions is the main goal of this matrix. To be more precise, the off-diagonal columns represent misclassifications into other classes, and the diagonal cells indicate the number of correct predictions matching the real class. The notable findings from Figure 4 are

- Diagonal Dominance: In every confusion matrix, the diagonal elements have greater values, indicating a majority number of accurate classifications.
- Misclassification Patterns: The off-diagonal elements of the matrix indicate distinct tendencies of misclassification that differ between the models. For example, certain models may more commonly misidentify specific types of lesions.
- Performance Improvement: The Max Voting method exhibits superior performance by minimizing misclassifications, hence highlighting the efficacy of ensemble techniques in enhancing classification accuracy.



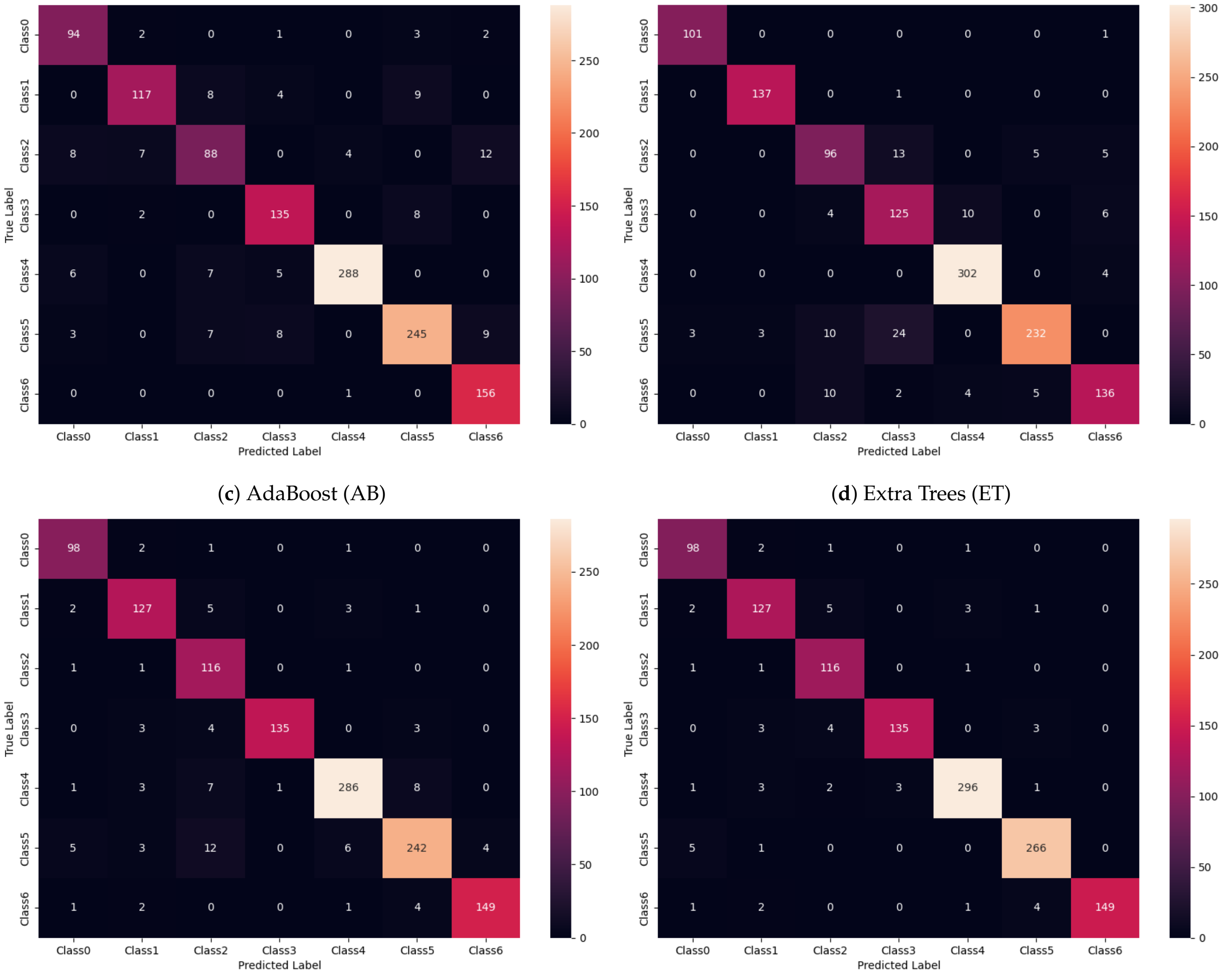
	Class0	Class1	Class2	Class3	Class4	Class5	Class6
Class0	77	9	0	1	0	7	8
Class1	0	94	19	0	25	0	0
Class2	0	29	58	0	17	0	15
Class3	0	0	16	109	0	20	0
Class4	0	0	9	12	285	0	0
Class5	16	0	12	0	0	244	0
Class6	0	0	0	2	1	0	154

(**a**) Random Forest (RF)

	Class0	Class1	Class2	Class3	Class4	Class5	Class6
Class0	89	0	5	0	8	0	0
Class1	12	105	0	4	0	1	16
Class2	9	0	76	8	4	13	9
Class3	7	2	0	128	4	0	4
Class4	0	4	0	0	290	8	4
Class5	0	14	0	0	21	237	0
Class6	3	7	2	0	0	2	143

(**b**) Gradient Boosting (GB)

Figure 4. *Cont.*



(**c**) AdaBoost (AB)

(**d**) Extra Trees (ET)

(**e**) CatBoost (CB)

(**f**) Max Voting (MV)

Figure 4. Confusion matrices of various models.

4.5. Performance of Multiple Models on the HAM10000 Dataset

The analysis and experimental results of our models applied to the HAM10000 dataset are shown in this part. The outcomes of five ensemble learning model types—Random Forest, Gradient Boosting, AdaBoost, CatBoost, and Extra Trees—were compared using this dataset's images. A batch size of 44 was used to train each model for a period of 20 epochs. Metrics like average accuracy, precision, recall, and F1-score were included in the results. Table 12 displays the outcomes of several models tested on the validation dataset, which includes 1293 images of seven different skin lesion types, after the training process was complete. According to Table 12, the other models were outperformed by the proposed Max Voting method when measuring the accuracy, precision, recall, and F1-score. The performance metrics for each class of the Max Voting method are shown in Table 13. Based on the observations, Class 5 has the highest level of precision, while Class 1 has the lowest.

Table 12. Performance of multiple models on the validation dataset.

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)	Key Features
Random Forest	82.41	79.53	78.35	78.77	Trees are trained independently.
Gradient Boosting	86.20	73.74	84.06	84.21	Trees are trained sequentially.
AdaBoost	90.64	88.88	89.64	89.14	Automatically handles missing values
Extra Trees	91.12	90.23	90.82	90.43	Uses the whole learning sample to construct trees
CatBoost	93.06	92.46	93.71	92.93	Specialized in dealing with categorical data
Max Voting	95.80	95.04	95.44	95.20	Combine the predictions from multiple models

Table 13. Class-wise performance metrics of Max Voting method.

Class	Avg. Precision (%)	Avg. Recall (%)	Avg. F1-Score (%)
Class 0	96.08	90.74	93.33
Class 1	92.03	91.37	91.70
Class 2	97.48	90.63	93.93
Class 3	93.10	97.83	95.41
Class 4	96.73	98.01	97.37
Class 5	97.79	96.73	97.26
Class 6	94.90	99.98	97.39

4.6. Performance of Multiple Models on the ISIC 2018 Dataset

Here, we present an extensive study and experimental results using several ensemble learning models on the 2018 International Skin Imaging Collaboration (ISIC) dataset. Among the models evaluated are Random Forest, AdaBoost, CatBoost, Gradient Boosting, and Extra Trees. Using parameters including average accuracy, recall, precision, and F1-score, we evaluated the models' performance following 32 epochs of training. This dataset includes 13,788 dermoscopy images of seven distinct skin lesion types: melanoma, basal cell carcinoma, actinic keratoses, benign keratosis-like lesions, dermatofibroma, and vascular lesions. Following training, the models were tested using a validation dataset that included 1500 images.

The ensemble learning models perform noticeably better with the ISIC 2018 dataset when the Max Voting method is applied, as the findings in Table 14 demonstrate. For accuracy, precision, recall, and F1-score, the Max Voting approach outperforms the standalone models such as CatBoost, Random Forest, Gradient Boosting, AdaBoost, and Extra Trees. Max Voting produced the most remarkable overall metrics, with 96.20% accuracy, 96.30% precision, and 95.50% recall.

The class-wise studies displayed in Table 15 show that the Max Voting method routinely demonstrates improved recall and accuracy for classifying various skin lesions, especially for identifying melanoma and melanocytic nevi. This technique produced the best precision (98.20%) for melanoma (Class 5) and the maximum recall for melanocytic nevi (Class 4). All things considered, the Max Voting method shows its value by raising the precision and dependability of skin cancer classification, which makes it a helpful instrument for early diagnosis and therapy and, in the end, improved patient outcomes.

Table 14. Multi-model performance on ISIC 2018 validation dataset.

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
Random Forest	84.50	82.30	81.20	81.74
Gradient Boosting	88.70	76.50	87.90	87.20
AdaBoost	92.10	90.50	91.30	90.90
Extra Trees	92.80	91.70	92.50	92.10
CatBoost	94.30	93.80	94.90	94.20
Max Voting	96.20	95.50	96.30	95.63

Table 15. ISIC 2018 dataset class-wise Max Voting performance metrics.

Class	Avg. Precision (%)	Avg. Recall (%)	Avg. F1-Score (%)
Class 0	97.10	91.20	94.10
Class 1	93.20	92.50	92.80
Class 2	96.30	91.70	94.80
Class 3	94.00	98.50	96.20
Class 4	97.50	98.70	98.10
Class 5	98.20	97.10	97.65
Class 6	95.80	96.00	97.36

4.7. Comparison of State-of-the-Art Models vs. Proposed Max Voting Model

Factors including the dataset, model architecture, and preprocessing methods used all have an impact on how well and how accurately ML models perform. Here, we compare our suggested Max Voting model to various state-of-the-art models that are used for skin cancer classification in detail. The recent studies in this area have shown varying degrees of success, but there is still a need for models that offer higher accuracy and reliability. The testing accuracy values for our proposed method and the existing approaches are summarized in Table 16. As illustrated, our proposed Max Voting model achieves a higher accuracy of 95.80%, surpassing the other models listed. In Table 16, we compare our proposed Max Voting model with three other models: a Convolutional Neural Network (CNN) as referenced by Gouda et al. [45], a Hybrid CNN as proposed by Khan et al. [46], and a Max Voting-CNN approach by Hossain et al. [47]. Each of these models employs different strategies to enhance the accuracy of skin cancer classification. CNN (Gouda [45]): the basic CNN model achieved an accuracy of 83.2%. CNNs are effective in image classification tasks due to their ability to learn spatial hierarchies, but they may not fully capture complex patterns in skin cancer images when used alone. Hybrid CNN (Khan [46]): this model integrates traditional CNNs with other techniques to improve the performance, reaching an accuracy of 84.5%. Hybrid approaches aim to combine the strengths of multiple methods but can still fall short in handling diverse skin cancer datasets. Max Voting-CNN (Hossain [47]): by using a Max Voting strategy within CNNs, this model achieved an accuracy of 93.18%. The Max Voting method helps to combine predictions from different CNN architectures, improving the overall accuracy. However, the reliance on CNNs alone may limit the potential gains.

Our approach, which employs a Max Voting strategy across multiple pre-trained ensemble models, achieves the highest accuracy of 95.80%. This method combines the predictions from diverse models such as AdaBoost, CatBoost, Random Forest, Gradient Boosting, and Extra Trees, effectively leveraging their individual strengths and mitigating their weaknesses. To further improve the model's performance, a genetic algorithm is used to choose the features optimally. Why the proposed Max Voting ensemble model is superior is explained by a number of facts. One is called Max Voting ensemble, which uses several model types to obtain additional data from the dataset. Secondly, the ensemble technique reduces the possibility of overfitting and improves the generalizability of the model. Lastly, the proposed Max Voting ensemble model is significantly more precise and trustworthy than the existing approaches for classifying skin cancer.

Table 16. Comparison of proposed model with state-of-the-art models.

Reference	Model	Accuracy (%)
[45] (2022)	CNN	83.2
[46] (2024)	Hybrid CNN	84.5
[47] (2023)	Max Voting-CNN	93.18
Proposed model	Max Voting Ensemble	95.80

5. Conclusions

This work proposes to use the Max Voting method together with sophisticated pre-trained ensemble models to identify and categorize skin cancer. The study demonstrates how the diagnosis and classification of skin cancer can be much improved by combining predictions from several models. At 95.80% accuracy, the Max Voting method beats standalone models such as AdaBoost, CatBoost, Random Forest, Gradient Boosting, and Extra Trees. Further demonstrated by the study are the effectiveness and high dependability of the Max Voting ensemble approach by using a genetic algorithm to produce the best feature vectors from image data, thus improving the input for the ensemble model and improving the classification results. The superior F1-measure, recall, and precision values of the approach indicate its potential as a reliable tool for skin cancer classification. The findings suggest that this method not only improves the accuracy and reliability of skin cancer detection but also establishes a new standard for future research in this area. Nevertheless, it is important to acknowledge certain constraints, such as the reliance on the dataset's quality and diversity, the computational complexity and resources needed for training and implementing ensemble models, and the understanding of ensemble models. The future research may investigate the incorporation of deep learning models, building skin cancer detection systems that operate in real-time, the examination of hybrid ensemble methods, further studies on advanced techniques for engineering and selecting features, improving the ability to explain and interpret ensemble models, and exploring the use of the Max Voting ensemble approach in other areas of medical imaging. By pursuing these research directions and addressing the noted limitations, the field can continue to advance, leading to more accurate, reliable, and efficient diagnostic tools that can significantly improve patient outcomes in dermatology and beyond.

Author Contributions: Methodology, P.N.; Investigation, P.N.; Writing—original draft, P.N.; Supervision, P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data were contained in the main text.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Apalla, Z.; Nashan, D.; Weller, R.B.; Castellsagué, X. Skin cancer: Epidemiology, disease burden, pathophysiology, diagnosis, and therapeutic approaches. *Dermatol. Ther.* **2017**, *7*, 5–19. [CrossRef] [PubMed]
2. Hu, W.; Fang, L.; Ni, R.; Zhang, H.; Pan, G. Changing trends in the disease burden of non-melanoma skin cancer globally from 1990 to 2019 and its predicted level in 25 years. *BMC Cancer* **2022**, *22*, 836. [CrossRef] [PubMed]
3. Zelin, E.; Zalaudek, I.; Agozzino, M.; Dianzani, C.; Dri, A.; Di Meo, N.; Giuffrida, R.; Marangi, G.F.; Neagu, N.; Persichetti, P.; et al. Neoadjuvant therapy for non-melanoma skin cancer: Updated therapeutic approaches for basal, squamous, and merkel cell carcinoma. *Curr. Treat. Options Oncol.* **2021**, *22*, 35. [CrossRef] [PubMed]

4. Magnus, K. The Nordic profile of skin cancer incidence. A comparative epidemiological study of the three main types of skin cancer. *Int. J. Cancer* **1991**, *47*, 12–19. [CrossRef]
5. Bhatt, H.; Shah, V.; Shah, K.; Shah, R.; Shah, M. State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: A comprehensive review. *Intell. Med.* **2023**, *3*, 180–190. [CrossRef]
6. Goyal, M.; Knackstedt, T.; Yan, S.; Hassanpour, S. Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities. *Comput. Biol. Med.* **2020**, *127*, 104065. [CrossRef] [PubMed]
7. Raval, D.; Undavia, J.N. A Comprehensive assessment of Convolutional Neural Networks for skin and oral cancer detection using medical images. *Healthc. Anal.* **2023**, *3*, 100199. [CrossRef]
8. Iqbal, S.N.; Qureshi, A.; Li, J.; Mahmood, T. On the analyses of medical images using traditional machine learning techniques and convolutional neural networks. *Arch. Comput. Methods Eng.* **2023**, *30*, 3173–3233. [CrossRef] [PubMed]
9. Elgamal, M. Automatic skin cancer images classification. *Int. J. Adv. Comput. Sci. Appl.* **2013**, *4*, 287–294. [CrossRef]
10. Kanca, E.; Ayas, S. Learning Hand-Crafted Features for K-NN based Skin Disease Classification. In Proceedings of the International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–4.
11. Bansal, P.; Vanjani, A.; Mehta, A.; Kavitha, J.C.; Kumar, S. Improving the classification accuracy of melanoma detection by performing feature selection using binary Harris hawks optimization algorithm. *Soft Comput.* **2022**, *26*, 8163–8181. [CrossRef]
12. Moradi, N.; Mahdavi-Amiri, N. Kernel sparse representation based model for skin lesions segmentation and classification. *Comput. Methods Programs Bio Med.* **2019**, *182*, 105038. [CrossRef] [PubMed]
13. Balaji, V.R.; Suganthi, S.T.; Rajadevi, R.; Krishna Kumar, V.; Saravana Balaji, B.; Pandiyan, S. Skin disease detection and segmentation using dynamic graph cut algorithm and classification through Naive Bayes classifier. *Measurement* **2020**, *163*, 107922. [CrossRef]
14. Brinker, T.J.; Hekler, A.; Utikal, J.S.; Grabe, N.; Schadendorf, D.; Klode, J.; Berking, C.; Steeb, T.; Enk, A.H.; Von Kalle, C. Skin cancer classification using convolutional neural networks: Systematic review. *J. Med. Internet Res.* **2018**, *20*, e11936. [CrossRef] [PubMed]
15. Shokouhifar, A.; Shokouhifar, M.; Sabbaghian, M.; Soltanian-Zadeh, H. Swarm intelligence empowered three-stage ensemble deep learning for arm volume measurement in patients with lymphedema. *Biomed. Signal Process. Control* **2023**, *85*, 105027. [CrossRef]
16. Bao, J.; Hou, Y.; Qin, L.; Zhi, R.; Wang, X.-M.; Shi, H.-B.; Sun, H.-Z.; Hu, C.-H.; Zhang, Y.-D. High-throughput precision MRI assessment with integrated stack-ensemble deep learning can enhance the preoperative prediction of prostate cancer Gleason grade. *Br. J. Cancer* **2023**, *128*, 1267–1277. [CrossRef] [PubMed]
17. Guergueb, T.; Akhloufi, M.A. Skin Cancer Detection using Ensemble Learning and Grouping of Deep Models. In Proceedings of the 19th International Conference on Content-Based Multimedia Indexing, Graz, Austria, 14–16 September 2022; pp. 121–125.
18. Avanija, J.; Reddy, C.C.; Reddy, C.S.; Reddy, D.H.; Narasimhulu, T.; Hardhik, N.V. Skin Cancer Detection using Ensemble Learning. In Proceedings of the International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 14–16 June 2023; IEEE: Piscataway, NJ, USA; pp. 184–189.
19. Tschandl, P.; Rosendahl, C.; Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data* **2018**, *5*, 180161. [CrossRef] [PubMed]
20. Jain, S.; Singhania, U.; Tripathy, B.; Nasr, E.A.; Aboudaif, M.K.; Kamrani, A.K. Deep learning-based transfer learning for classification of skin cancer. *Sensors* **2021**, *21*, 8142. [CrossRef]
21. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
22. Ibraheem, N.A.; Hasan, M.M.; Khan, R.Z.; Mishra, P.K. Understanding color models: A review. *ARPN J. Sci. Technol.* **2012**, *2*, 265–275.
23. Sharma, B.; Rupali Nayyer, R. Use and analysis of color models in image processing. *J. Food Process. Technol.* **2016**, *7*, 533. [CrossRef]
24. Nguyen, T.P.; Vu, N.-S.; Manzanera, A. Statistical binary patterns for rotational invariant texture classification. *Neurocomputing* **2016**, *173*, 1565–1577. [CrossRef]
25. Song, T.; Feng, J.; Wang, S.; Xie, Y. Spatially weighted order binary pattern for color texture classification. *Expert Syst. Appl.* **2020**, *147*, 113167. [CrossRef]
26. Liu, L.; Chen, J.; Fieguth, P.; Zhao, G.; Chellappa, R.; Pietikäinen, M. From BoW to CNN: Two decades of texture representation for texture classification. *Int. J. Comput. Vis.* **2019**, *127*, 74–109. [CrossRef]
27. Hong, H.; Zheng, L.; Pan, S. Computation of Gray Level Co-Occurrence Matrix Based on CUDA and Optimization for Medical Computer Vision Application. *IEEE Access* **2018**, *6*, 67762–67770. [CrossRef]
28. Shukla, A.K. Simultaneously feature selection and parameters optimization by teaching–learning and genetic algorithms for diagnosis of breast cancer. *Int. J. Data Sci. Anal.* **2024**, 1–22. [CrossRef]
29. Ngo, G.; Beard, R.; Chandra, R. Evolutionary bagging for ensemble learning. *Neurocomputing* **2022**, *510*, 1–14. [CrossRef]
30. Chen, W.; Lei, X.; Chakrabortty, R.; Chandra Pal, S.; Sahana, M.; Janizadeh, S. Evaluation of different boosting ensemble machine learning models and novel deep learning and boosting framework for head-cut gully erosion susceptibility. *J. Environ. Manag.* **2021**, *284*, 112015. [CrossRef]

31. Lin, W.; Wu, Z.; Lin, L.; Wen, A.; Li, J. An ensemble random forest algorithm for insurance big data analysis. *IEEE Access* **2017**, *5*, 16568–16575. [CrossRef]
32. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967. [CrossRef]
33. Mienye, I.D.; Yanxia Sun, Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access* **2022**, *10*, 99129–99149. [CrossRef]
34. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. *J. Big Data* **2020**, *7*, 94. [CrossRef]
35. Sharma, D.; Kumar, R.; Jain, A. Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning. *Meas. Sens.* **2022**, *24*, 100560. [CrossRef]
36. Dogan, A.; Birant, D. A weighted majority voting ensemble approach for classification. In Proceedings of the 2019 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, 11–15 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
37. Backes, A.R.; Casanova, D.; Bruno, O.M. Color texture analysis based on fractal descriptors. *Pattern Recognit.* **2012**, *45*, 1984–1992. [CrossRef]
38. Bianconi, F.; Harvey, R.; Southam, P.; Fernández, A. Theoretical and experimental comparison of different approaches for color texture classification. *J. Electron. Imaging* **2011**, *20*, 043006. [CrossRef]
39. Palm, C. Color texture classification by integrative co-occurrence matrices. *Pattern Recognit.* **2004**, *37*, 965–976. [CrossRef]
40. Humeau-Heurtier, A. Texture feature extraction methods: A survey. *IEEE Access* **2019**, *7*, 8975–9000. [CrossRef]
41. Liu, L.; Fieguth, P.; Guo, Y.; Wang, X.; Pietikäinen, M. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognit.* **2017**, *62*, 135–160. [CrossRef]
42. Qi, X.; Zhao, G.; Shen, L.;Li, Q.; Pietikäinen, M. LOAD: Local orientation adaptive descriptor for texture and material classification. *Neurocomputing* **2016**, *184*, 28–35. [CrossRef]
43. Tohka, J.; Van Gils, M. Evaluation of machine learning algorithms for health and wellness applications: A tutorial. *Comput. Biol. Med.* **2021**, *132*, 104324. [CrossRef]
44. Heydarian, M.; Doyle, T.E.; Samavi, R. MLCM: Multi-label confusion matrix. *IEEE Access* **2022**, *10*, 19083–19095. [CrossRef]
45. Gouda, W.; Sama, N.U.; Al-waakid, G.; Humayun, M.; Jhanjhi, N.Z. Detection of Skin-Cancer Based on Skin Lesion Images Using Deep Learning. *Healthcare* **2022**, *10*, 1183. [CrossRef]
46. Khan, M.A.; Muhammad, K.; Sharif, M.; Akram, T.; Kadry, S. Intelligent fusion-assisted skin lesion localization and classification for smart healthcare. *Neural Comput. Appl.* **2024**, *36*, 37–52. [CrossRef]
47. Hossain, M.M.; Hossain, M.M.; Arefin, M.B.; Akhtar, F.; Blake, J. Combining State-of-the-Art Pre-Trained Deep Learning Models: A Noble Approach for Skin Cancer Detection Using Max Voting Ensemble. *Diagnostics* **2023**, *14*, 89. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.